A new Method on Resource Scheduling in grid systems based on Hierarchical Stochastic Petri net

Mohammad Shojafar
Msc. Computer Eng. Computer & Elec. Department
Islamic Azad Universty of Qazvin
Qazvin,Iran
Shojafar@qiau.ac.ir

Siamak Barzegar
Msc. Computer Eng. Computer & Elec. Department
Islamic Azad University of Qazvin
Qazvin,Iran
Barzegar@qiau.ac.ir

Mohammad Reza Meibody
PHD in Computer Science, Computer Department
Amirkabir University
Tehran,Iran
Mmeybodi@aut.ac.ir

***Abstract*—one of the main goals of Grid systems is to share system resources and schedule resource requests. The resource scheduling problem becomes complex when resources are distributed, heterogeneous, dynamic and autonomous. In this paper, we have proposed a new method to optimize the resource Scheduling in grid computing based on categorized requests in three layers which depend on Hierarchical Stochastic Petri net model called HSPN. These layers are Home, Local and Grid layers. We have divided our tasks in these layers. Each layer has special function to receive subtask and deliver data to up/down layer. Then, we have compare HSPN with other resource scheduling such as Min.min and Max.min algorithms. These results show that the HSPN algorithm provides better results than Max.min but the results are weaker than Min.min algorithm in the same conditions.**

Keywords-Grid Computing; Hierarchical Stochastic Petri Net; Resource Scheduling; Resource Allocation; Modeling

I. INTRODUCTION

In recent years, Grid computing application development has increased. In fact, a Grid computing system is for heterogeneous communication systems to share and use resources in other ones. So, Grid computing in which a network of computers is integrated to create a very fast virtual computer becomes more prevalent [1].

Grid Computing has evolved from its roots in science academia and it is currently at the onset of mainstream commercial adoption. But with the recent explosion of commercial interest in grid, we are seeing some industry confusion about what the term means. Grid computing is the combination of computer resources from multiple administrative Domains for ordinary goals [2].

Grid computing is applying the resources of many computers in a network to a single problem at the same time to a scientific or technical problem that requires great number of computer processing cycles or access to large amounts of data. Many different groups get to gather in this collaborative effort known as Virtual Organizations (VOs) [3].

These VOs may be formed to solve a single task and may then disappear just quickly [4]. One of the promises of Grid Computing is to enable the execution of applications across multiple Sites. Some of these applications require coordinated access to resources managed by autonomous entities. This coordinated access is known as resource Scheduling. Grid computing technology aims to realize the global resource sharing in the distributed, heterogeneous and autonomic environment and assemble these resources to meet the requirements of the applications. With the rapid development of the grid computing technology, many people began to run large scale parallel applications on grids [5].

Different ways presented to model different parts of the Grid computing. Because the amount of dynamic Grid computing multiple request has sent too much, requires are used to evaluate the ability to model parallel and concurrent systems. Petri net is one of the tools work in ways. One of the simplest and most flexible methods is Petri net.

Stochastic Petri Nets are a modeling formalism that can be conveniently used for the analysis of complex models of Discrete Event Dynamic Systems (DEDS) and for their performance and reliability evaluation. The automatic construction of the probabilistic models relies on a set of results that underlay the dynamic behaviors of these nets which derive the theory of untimed Petri nets [6].

In this paper, in the latter chapter, we talk about the related works on resource Scheduling on grid computing based on Petri net. In Section 3, the approach is introduced. Here, we have introduced HSPN by some figures clearly. Performance evaluation and comparing proposed HSPN algorithm with min.min and Max.min Algorithms are presented in section 4. Finally, as a conclusion part or conclusion section, advantages and disadvantages and future works are given.

978-1-4244-7224-6 /10/$26.00

II. RELATED WORKS

Grid Computing is one of the most important issues in the Grid, in quality of allocating and scheduling resources in the system. From past up to present, scientists and researchers have had lot of efforts in network science and Grid computing.

Several works have been done in the field of Grid computing. Yu and Buyya done [7, 8] a lot of research on Grid workflow in 2005 and 2006. They provide a general framework and structure that are different categories that make it possible to facilitate workflow in Grid system. This model is based on DAG scheduling algorithms ability to model several researches. Their focus is on the Workflow design that the system manages and algorithms schedule. DAG is limited to model dynamic systems with far better features. But, being a complex model, Grid environments is not accountable to the needs and changes. Therefore, researchers think about following the use of modeling tools that concurrency and synchronization are more accountable in applications with speed and accuracy. Hence, Petri net family such as CPN (Colored Petri net), SPN (Stochastic Petri net) are effective tools in scheduling and resource allocation in emerged Grid.

In reference [9] three Layer model is presented based on hierarchical time Petri net (HTPN) in 2005. This model is based on different Petri net models for Grid Scheduler. Three layers in this paper are Local Scheduler, Home Scheduler and Grid Scheduler. This idea is focused on independent task and it is not considered as dependent tasks. Dependent tasks scheduling is presented in reference [10] in Extended Time Petri net in 2006. Grid application scheduling model based on Petri net is extended in reference [11] in 2007. This paper has listed all works on scheduling in grid by the help of Petri net families. It has presented scheduling algorithms based on four-level scheduling. It considers the independent tasks, clearly. Here, we have presented algorithms based on Hierarchical stochastic Petri net. In this paper, we have introduced a method for scheduling and allocating resources by the help of SPN properties.

III. APPROACH: RESOURCE SCHEDULING ON HIERARCHICAL STOCHASTIC PETRI NET MODEL (HSPN)

As we know, Resource Scheduling in grid system is very important. Based on the hierarchical scheme and distributed scheme given in reference [12], we have given a three-level scheduling scheme shown in Figure 1. In three-level scheduling scheme, the resources are connected via a three-level hierarchical network. The first level is a wide-area network (WAN) that connects local area networks (LANs). The second level is a LANs that connects computing resources (personal computers and high performance computers), stored resources and other resources at the third-level.

Unlike hierarchical scheme and distributed scheme, all tasks are submitted to home scheduler instead of grid scheduler at their own sites in the three-level scheduling scheme, which shows the autonomy of grid resource and is convenient for user to submit and supervise tasks. In addition, the three-level scheduling scheme adds local scheduler between home scheduler and grid scheduler, which not only lightens the pressure of grid scheduler, but also makes tasks be possibly executed in local area. Fig. 1 illustrates this Hierarchical model.

We have modeled these schemes in hierarchical stochastic Petri net. Each task is submitted to its home scheduler by user with its processing requirements such as estimated processing time, estimated communication time, deadline and parallel degree. The process of task submission and assignment is as follows.

- A task is submitted to home scheduler by user through home machine. The home scheduler analyzes the submitted task. If the task is completed within its deadline in home machine, then the task will be executed on the home machine. Otherwise, the task is sent to the local scheduler.
- When local scheduler receives the task submitted by home scheduler, it decides whether the task can be completed within its deadline in the local area network. If so, the local scheduler assigns subtasks of the task to some machines in the local area network according to some algorithm. Otherwise, the task is sent to the grid scheduler.
- When grid scheduler receives the task submitted by local scheduler, it inserts the task into the queue of tasks. Table 1 illustrates the notation of HSPN Model. Table 1 illustrates the Places and Transition for simulating our hierarchical scheduling.

The HSPN model of our Approach is illustrated in Fig. 2.

We have implemented this model on resource Scheduling simulator in grid system like Gridsim. So, at first we have introduced some definitions to implement this HSPN model in Gridsim software for resource Scheduling.

In GridSim, entities use events for both service requests and service deliveries. The events can be raised by any entity to be delivered immediately or with specified delay to itself or other entities. The events that are originated from the same entity are called *internal events* and those which are originated from the external entities are called *external events*. The GridSim protocols are used to define entity services.

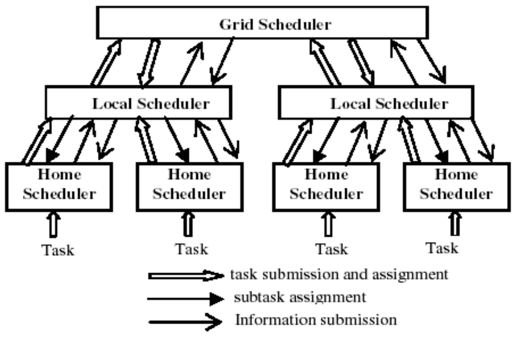

Figure 1. Three Layer for assigning task and sub task

TABLE I. NOTATION FOR HSPN MODEL

Places(P)	Description	Transition (T)	Description
P_1	The Task is Submitted By USER	t_0	Submitting task to home machine
P_2	Resources	t_1	Getting task and machine info.
P_3	Scheduling Task	t_2	Cannot complete task within deadline
P_4	Submitting Task	t_3	Can complete task within deadline
P_5	Executing Task	t_4	Sending task to local scheduler
P_6	Executing Task in Remote	t_5	Assigning task to home scheduler
P_7	Task Completed	t_6 or t_8	Executing remote or home task
P_8	Remote Task Completed	t_7	Submitting completed remote subtask
Pk_1,PK_3,PK_5	Output Parts: Submitting Task, Completing subtask, Machine info.	t_9	Return completed task
Pk_2,PK_4	Input Parts(Assign): Remote Task, Task Completed	t_{10}	Providing machine information

An event is called *synchronous* when the event source entity waits until the event destination entity performs all the actions associated with the event (i.e., the delivery of full service). An event is called *asynchronous* when the event source entity raises an event and continues with other activities without waiting for their completion. When the destination entity receives such events or service requests, it responds back which is resulted by sending one or more events, which can then take appropriate actions.

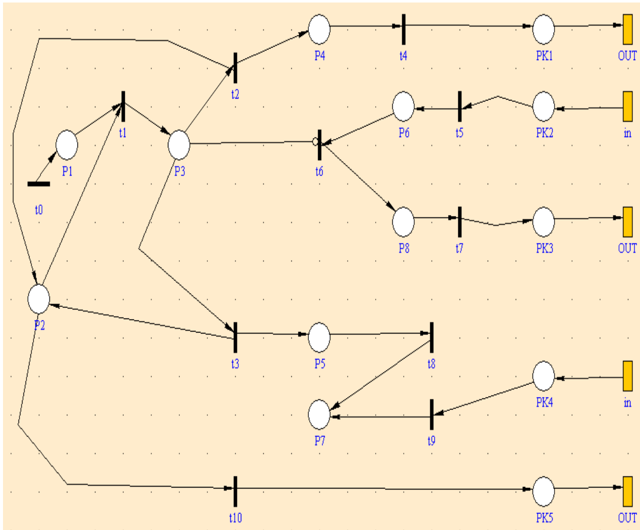

Figure 2. HSPN model for Resource scheduling

It should be noted that external events can be synchronous or asynchronous, but internal events need to be raised as asynchronous events only to avoid deadlocks [13].

IV. PERFORMANCE EVALUATION IN HSPN

A proposed Idea is an important part of related works done in this field that is assessed before.

We consider uniform and equal parameters for algorithms Min-min, Max-min in compare with HSPN and then evaluate it. So, to evaluate, first, we need to implement this approach on grids.

Here, we suppose that we have 4 machines, and each one has 4 processors and 1000 Giga byte information, then each processor has the same resources. It means that the processors for each machine are homogenous. Each machine in each site is different from the other machines in the other sites.

So, each site is heterogeneous in compare with the others. There are 100 tasks that come from each machine, each task has special time and cost. We have considered our approach to simulate resource Scheduling depend on cost and time. If we use more than hundred tasks for more than 100 users, we will not have enough space for heap in Java for grids. So, we use just 100 users, here, our tasks are separated in 4 parts as follows in Table 2. Each 25 request for each group is ordered from thin request to thick request. Thin request is a request that it is easy to response and do not have more resource. Almost resource needing is less than 3 Resources in our simulation.

TABLE II. COST/TIME CONSIDERATION FOR 100 TASK ASSUMED IN SIMULATION AND TESTING

Cost Consider	Time Consider	Number of tasks Consider
No	No	25-First
No	Yes	25-Second
Yes	No	25-Third
Yes	Yes	25-Forth

Thick request is a request that needs resources more than 8. These requests should be often searched by the grid scheduler.

As mentioned before, we should first search Home machine, then its Database. Searching Schedule in each machine is called Home schedule. If we do not find our required resource of cost and time, then we will send request to coordinator to find slightly resource on local site. This scheduler is called the local scheduler. After that, if we do not find our resource depend on our parameters, we will send the request to broker to find different sites. In this stage it is named Grid scheduler. We compare our approach with other algorithms. The compared algorithms are:

1) *Min-min:*

In Min-min, the minimum completion time for each task is computed respecting to all machines. The task with the overall minimum completion time is selected and assigned to the corresponding machine. The newly mapped task is removed and the process repeats until all tasks are mapped [14, 15, 16, 17].

2) *Max.min:*

The Max min heuristic is very similar to the Min-min algorithm. The set of minimum a completion time is calculated for every task. The task with overall maximum completion time from the set is selected and assigned to the corresponding machine [14, 15, 16, and 17].

3) *Our approach (HSPN):*

This approach is depending on hierarchical scheduling.

We have considered some important parameters in performance evaluation. These parameters focused on are listed below:

- Average Response Time/Per Request(ART)

The important parameter to evaluate interval between receiving requests and sending answers is the Response time. Response time is important on the web for sending/receiving tasks. Average response time in HSPN that means the summation time for passing these three layers depends on each request. There might be some tasks that cannot be responded in their home and local schedule time. So, the broker should find the nearest resource depends on its parameters. Here, we have considered this parameter for each task. We have computed this time for all 3 algorithms.

- Average Response Cost/Per Request(ARC)

Response cost is so important on the web for tasks. The delivered data cost or resource cost for task is the Response cost. All resource Scheduling algorithms consider this parameter. These algorithms almost consider the resource which is closer to the task site and the costs are less than the cost for each task. But, it may be some resources whose cost is slightly more than the needed cost considered here.

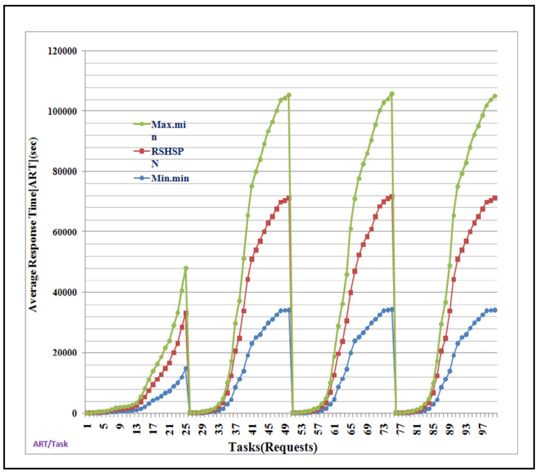

Figure 3. Average Response Time (ART) per request for Min.min/Max.min/RSHSPN

For example, if task1 finds a resource less than \$300, we get it \$300+ ((1/100)* 300) about \$303. It means we are looking for resources that cost less than 303\$. Average Response Cost is the cost that schedules find and the respond is accepted by users. Average response cost in HSPN means the costs summation for passing these three layers depend on each request. Some requests do not need to go to grid layer or local layer. So, their cost becomes less than other methods.

Figure 3 illustrates Average Response Time (ART) per requested time which comes from each user. This plot shows Min.min is better than max.min and HSPN in 25 first tasks and HSPN is between Min.min and Max.min. We have considered the maximum time of 36000 seconds and maximum cost of \$.3000. the time and cost is increased from 26th request up to 100.As you see, HSPN in ART for time and cost is near Min.min and sometimes in thick request, it is better than Min.min.

Our approach in ART is about 11-12% decreased time toward Max.min and in 4th class of the 25th task group, the thick requests are better than Min.min algorithms. Because in searching grid category, some tasks and resource findings get better results.

Figure 4 illustrates Average Response Cost (ARC) requested cost which comes from each user This plot shows that in the first 25th tasks, because cost and time are not mentioned, all three algorithms deliver each task in high value, but in the tasks second group, we haved etermined maximum cost up to $3000. So, Scheduling should be just the machines whose cost is less than the cost of each task considered. This consideration is (Cost (Task) + 1%*Cost (Task)). As we can see, HSPN decreases the cost with bounding the resources but this decrease is not better than Min.min algorithm. However, it is better than Max.min algorithm. In thick requests, this algorithm can act similar to Min.min and sometimes it is better than Min.min.

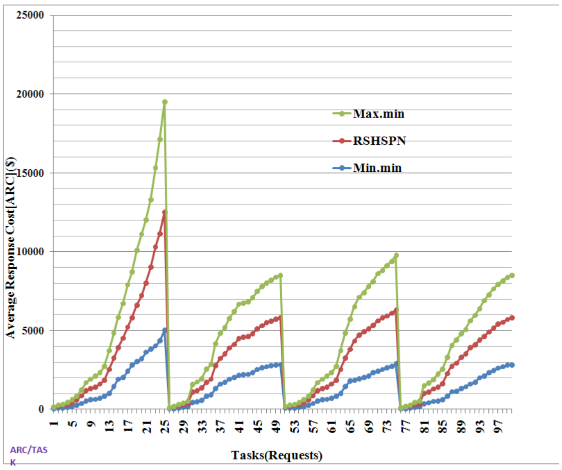

Figure 4. Average Response Cost (ARC) per request for Min.min/Max.min/RSHSPN

V. CONCLUSION

There is no optimal method to allocate resource and schedule the grid systems, because these systems are heterogeneous, dynamic and expansion in the world.

In this paper, a resource scheduling scheme is presented with three-level in grid computing environments. The resource scheduling scheme is modeled and analyzed by hierarchical Stochastic Petri net. Different Stochastic Petri net models are given in this paper to schedule the resource of different layers (home scheduler, local scheduler and grid scheduler). We have compared HSPN with two major Resource Scheduling algorithms. Min.min and Max.min algorithms are applicable algorithms in grid systems. HSPN is a method which works based on SPN in grid. This method is run, if the network content increases, the tasks and requests will increase in the network.

The experiments we have tested show that grid scheduler is the last infrastructure to search and assign resources into tasks or sub tasks.

REFERENCES

[1] G. Koole and R. Righter, “Resource Scheduling in Grid Computing”, Journal of Scheduling, Vol.11, 2008, pp.163-173.

[2] I. Foster,” What is the Grid? A Three Point Checklist”, Argonne National Laboratory & University of Chicago, Vol.1, Issue 6, 2002, pp.32-36.

http://www.mcs.anl.gov/~itf/Articles/WhatIsTheGrid.pdf

[3] I. Foster, C. Kesselman, S. Tuecke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing, ISBN: 3-540-42495-4, Lecture Notes In Computer Science, Vol. 2150, Springer-Verlag, London, 2001, pp. 1-4.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.129.8850&rep=rep1&type=pdf

[4] F. Berman, G. Fox, A. J.G. Hey, Grid Computing: Making the Global Infrastructure a Reality , ISBN: 978-0-470-85319-1, Vol. 1, Wiley, 2003.http://as.wiley.com/WileyCDA/WileyTitle/productCd-0470853190.html

[5] Xi. WEI, Zh. DING, Sh. XING, Ya. YUAN , Wi. W. LI,"VJM: A Novel Grid Resource Co-Scheduling Model for Parallel Jobs", International Journal of Grid and Distributed Computing, ISBN: 978-1-4244-3430-5, Vol. 1, 2009, No.2, pp.7-13.

http://ieeexplore.ieee.org/articleSale/Sarticle.jsp?arnumber=4813625

[6] G. Balbo,” Introduction to stochastic Petri nets”, Springer Lectures on Formal Methods and Performance Analysis, ISBN: 3-540-42479-2, 2002, pp. 84 - 155.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.1476&rep=rep1&type=pdf

[7] J. Yu and R. Buyya, “A taxonomy of Scientific Workflow Systems for Grid Computing”, Special Issue on scientific Workflows, SIGMOD Records, ISSN:0163-5808, Vol. 34, Issue 3, 2005, pp. 44-49. http://www.buyya.com/papers/workflow-sigmod05.pdf

[8] J. Yu and R. Buyya, ”A budget Constrained Scheduling of workflow Application on Utility Grids using Genetic Algorithms”, Workshop on Workflows in Support of Large-Scale Science, Proceedings High Performance Distributed Computing Paris, France, ISSN:1058-9244,Vol. 14, Issue 3,4, 2006, pp. 217-230.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.63.7273&rep=rep1&type=pdf

[9] H. Yaojun, J. Changjun, L. Xuemei, “Resource Scheduling Model for Grid Computing Based on Sharing Synthesis of Petri net”, Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, CSCWD 2005, UK, Vol. 1, 2005, pp. 367-372.

http://doi.ieeecomputersociety.org/10.1109/CSCWD.2005.194199

[10] H. Yaojun and L. Xuemei, “Modeling and Performance Analysis of Grid Task Scheduling Based on Composition and Reduction of Petri nets”, Proceedings of the 5th International Conference on Grid and Cooperative Computing (GCC’06), ISBN: 0-7695-2694-2, 2006, pp. 331-334. http://doi.ieeecomputersociety.org/10.1109/GCC.2006.62

[11] X. Zhao, B. Wang, Li. Xu, “Grid Application Scheduling Model Based on Petri Net with Changeable Structure”, Proceeding of 6th International Conference on Grid and Cooperative Computing (GCC 2007), ISBN: 0-7695-2871-6, 2007, pp. 733-736.

http://doi.ieeecomputersociety.org/10.1109/GCC.2007.71

[12] V. Subramani, R. Kettimuthu, S. Srinivasan, P. Sadayappan, “Distributed Job Scheduling on Computational Grids Using Multiple Simultaneous Requests”, Proceedings of 11th IEEE International Symposium on High Performance Distributed Computing (HPDC'02), ISBN: 0-7695-1686-6, Edinburgh, Scotland, 2002, pp. 359-366.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.8.6401&rep=rep1&type=pdf

[13] R. Buyya and M. Murshed,” GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing”, Journal of Concurrency and Computation: Practice and Experience (CCPE), Vol. 14, 2000, No. 13, pp.1175-1220.

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.60.8382&rep=rep1&type=pdf

[14] B. SenthilKumar, P. Chitra, G. Prakash, “Robust Task Scheduling on Heterogeneous Computing Systems using Segmented MaxR-

MinCT", International Journal of Recent Trends in Engineering, Vol. 1, May 2009, No.2, pp.63-65.

http://www.academypublisher.com/ijrte/vol01/no02/ijrte0102063065.pdf

[15] WU. Min-you, SH. Wei., "Segmented Min-Min: a static mapping algorithm for meta-tasks on heterogeneous computing systems ", Proceedings of 9th Heterogeneous Computing Workshop (HCW 2000) ['C], 2000, pp.375 –385.

http://meseec.ce.rit.edu/eecc722-fall2002/papers/hc/4/00843759.pdf

[16] T. D. Braun, H. J. Siegel, et al. "A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing system l-J", Journal of Parallel and Distributed Computing, Vol. 61 , Issue 6, ISSN:0743-7315, 2001, pp. 810-837. DOI: http://dx.doi.org/10.1006/jpdc.2000.1714

[17] R. Armstrong, D. Hensgen, T. Kidd, "The relative performance of various mapping algorithms is independent of sizable variances in run-time predictions", Proceedings of 7th IEEE Heterogeneous Computing Workshop (HCW '98) [C], 1998, pp.79-87.

DOI:http://dx.doi.org/10.1109/HCW.1998.666547